RAPPORT DE RECHERCHE EN INTELLIGENCE ARTIFICIELLE

UNIVERSITE PARIS VIII - GROUPE II, EQUIPE 9 - 1974-1975

En 1974-1975, le groupe se compose des membres suivants :

Jacques	ARVEILLER	Chargé de cours
Louis	AUDOIRE	Technicien
Slimane	BEN HAMOUDA	Etudiant
Jérôme	CHAILLOUX	Chargé de cours
Christian	COLERE	Technicien
Yves	DEVILLERS	Assistant
André	DUCAMP	Professeur
Michel	FLIESS	Maître de Conférences
Patrick	GREUSSAY	Assistant
Jaime	LOPEZ-KRAHE	Etudiant
Pierre-Louis	NEUMANN	Chargé de cours
Jean-Eric	SCHOETTL	Assistant
Roger	TANGUY	Technicien
Harald	WERTZ	Chargé de cours

En 1973-1974, l'Equipe de Recherche en Intelligence Artificielle de l'Université PARIS VIII s'est consacrée aux 4 thèmes principaux que voici :

1) les langages et les interprètes ;

2) les programmes de jeu ;

3) le dialogue homme-ordinateur en langue naturelle ;

4) les périphériques de visualisation et les périphériques sonores.

Le thème 1 est une des préoccupations constantes de l'Equipe : se doter d'outils de programmation puissants spécialement adaptés aux mini-ordinateurs. Les études sur LISP et ses langages dérivés se sont particulièrement développées cette année. De plus, Slimane BEN HAMOUDA met actuellement au point un interpréteur du langage ALGOL 68 sur ordinateur T 1600. Pierre-Louis NEUMANN a mis au point un interpréteur de PS : le système de productions de Newell et Simon.

Le thème 2 est une nouveauté dans les activités du groupe : nous le considérons comme un *BANC D'ESSAI* de techniques spécifiques d'Intelligence Artificielle : gestion d'arborescences, représentation de *SAVOIR* sur un domaine spécifique, traitement syntaxique et sémantique d'énoncés en langue naturelle.

A terme nous espérons construire des programmes capables de "lire" un traité d'échecs. Notre problème principal est actuellement de repérer quel est le savoir implicite et les présupposés nécessaires pour qu'une telle lecture soit possible.

Le thème 3 a pris cette année un nouvel essort grace à la mise en place sur l'ordinateur T 1600 du système MILISY de T. Moran.

Le thème 4 a été une reconduction des projets de l'année dernière sur les périphériques de visualisation. Jérôme CHAILLOUX a de surcroît mis en place un nouveau système : KWRTH destiné à faciliter l'accès aux musiciens du synthétiseur de sons connecté à la CAB 500.

Par ailleurs Jaime LOPEZ-KRAHE travaille depuis quelques mois sur le difficile problème de la reconnaissance automatique de caractères imprimés.

Le Groupe édite un nouveau périodique : le RIPERT BULLETIN et a édité un numéro en langue anglaise consacré à l'informatique musicale et à la synthèse de sons.

Au cours de cette année de recherche Jean-Eric SCHOETTL a brillamment soutenu une thèse de spécialité portant sur : *UN MODELE MATHEMATIQUE DE POSITIONNEMENT* .

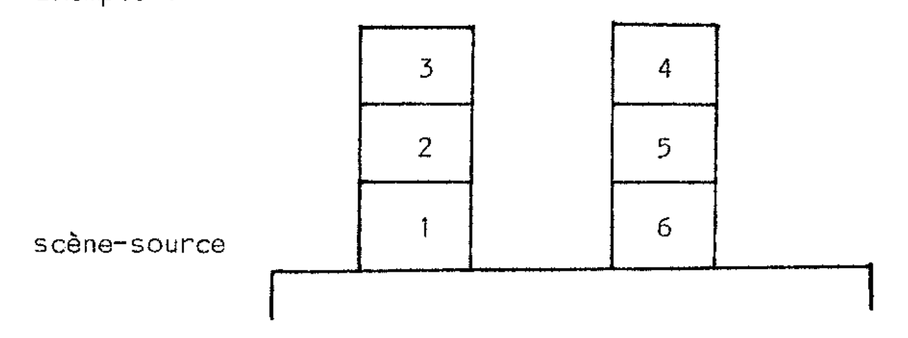
NOUVELLES IMPLEMENTATIONS DE LISP

Patrick GREUSSAY a réalisé sur l'ordinateur T 1600 une nouvelle version du LISP 510. La portabilité entre mini-ordinateurs soulignée dans le rapport de recherche 1973-1974 est confirmée par la rapidité des transferts d'une machine à l'autre : environ 3 semaines ont été nécessaires pour la nouvelle implémentation. Le système est complètement compatible avec le LISP CAE 510 et contient quelques fonctions supplémentaires. Le système comprend à l'heure actuelle :

1) l'interprète-noyau ;
2) un assembleur LAP donnant accès en LISP à toutes les instructions-machine de la T 1600 ;
3) un compilateur ;
4) plusieurs fonctions de TRACE :
5) une version restreinte de l'éditeur INTERLISP ;
6) une version restreinte du langage CONNIVER.

Voici, en ce qui concerne CONNIVER quelques exemples d'applications.

Le programme qui suit doit effectuer des manipulations sur la description symbolique d'une scène de bloc-ville.
Exemple :



Le problème posé est :

Etant donné la possibilité, en simulant le main d'un robot, de manipuler les blocs

- 1) un à un

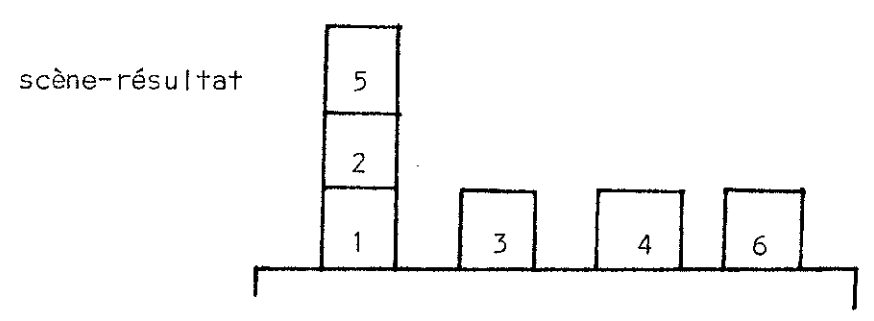
- 2) et seulement si aucun autre n'est posé dessus

générer un plan permettant d'obtenir à partir d'une scène-source, une scène-résultat.

Exemple :

telle que le bloc 5 soit posé sur le bloc 2.

Le plan généré devra se présenter sous la forme d'un programme exécutable en CONNIVER.



Voici le programme, le plan généré et son exécution.

```
14
0
(SETQ CONTEXT NIL)
NIL

(MAPC '((ON 1 TABLE) (ON 2 1) (ON 3 2)
        (ON 4 5) (ON 5 6) (ON 6 TABLE)) 'ADD)
NIL

(DE FREE (X ;; Z)
     (AND (PRESENT '(ON %Z ,X))
          (FREE1 Z)))
FREE

(DE FREE1 (X)
     (FREE X)
     (ADD '(ON ,X TABLE)))
FREE1

(DE SCENE (;; X Y)
     (FOR-EACH (ON %X %Y) (PRINT CURRENT)))
SCENE

(ADD (IF-ADDED IF-ON (ON %X %Y)
     %AUX% (X Y Z)
     (REMOVE (PRESENT '(ON ,X %Z)))))
IF-ON

(ADD (IF-ADDED IF-P-ON (IF-PUT-ON %X %Y)
     %AUX% (X Y)
     (FREE X)
     (FREE Y)
     (ADD '(ON ,X ,Y))))
IF-P-ON

(ADD (IF-REMOVED RMVD(ON %X %Z)
     %AUX% (X Z)
     (PUSH-PLAN 'REMOVE =(ON ,X ,Z))
     (PUSH-PLAN 'ADD =(ON ,X ,Y))))
RMVD

(DE PUSH-PLAN (INDIC WHAT)
     (SETQ PLAN (NCONC1 PLAN
                        (LIST INDIC (LIST 'QUOTE WHAT)))))
PUSH-PLA
```

```
;       ****** ****** ****** A C T I O N ****** ****** ******;
(SETQ PLAN NIL)
 NIL

(ASSUMING (IF-PUT-ON 5 2) (SCENE))
 (ON 5 2)
 (ON 3 TABLE)
 (ON 4 TABLE)
 (ON 6 TABLE)
 (ON 2 1)
 (ON 1 TABLE)
 T

(SCENE)
 (ON 6 TABLE)
 (ON 5 6)



 (ON 4 5)
 (ON 3 2)
 (ON 2 1)
 (ON 1 TABLE)
NIL

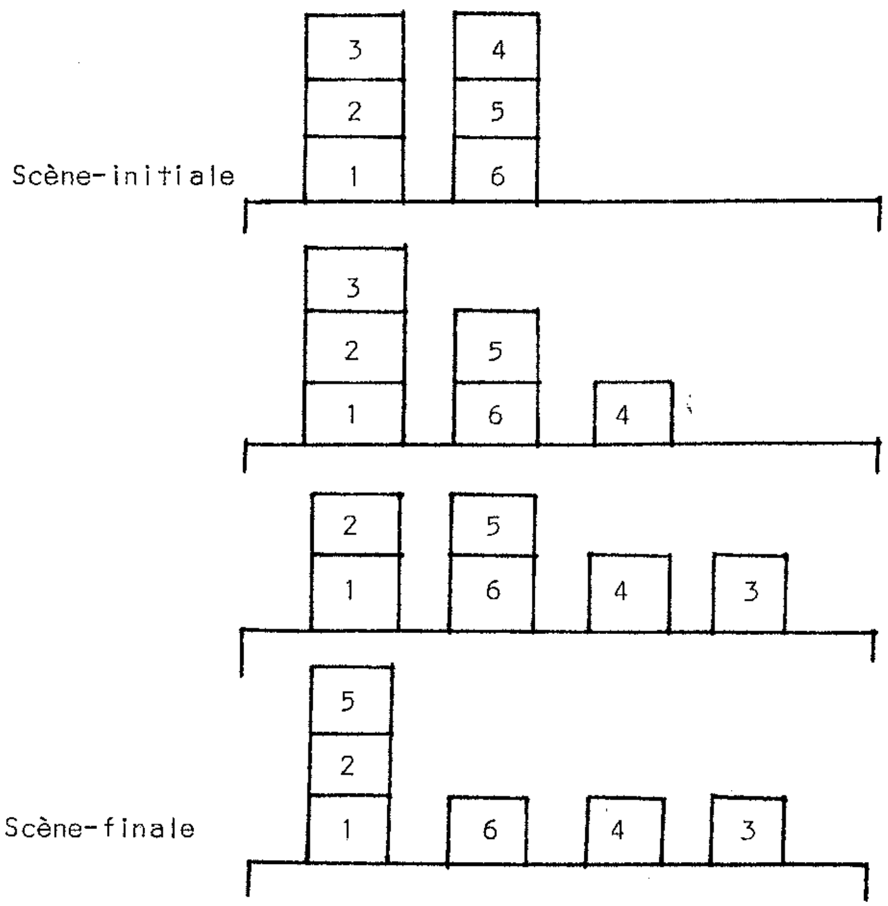
(MAPC PLAN 'PRINT)
 (REMOVE (QUOTE (ON 4 5)))
 (ADD (QUOTE (ON 4 TABLE)))
 (REMOVE (QUOTE (ON 3 2)))
 (ADD (QUOTE (ON 3 TABLE)))
 (REMOVE (QUOTE (ON 5 6)))
 (ADD (QUOTE (ON 5 2)))
NIL

;       ****** ****** EXECUTION DU PLAN ****** ****** ;
(MAPC '(IF-ON RMVD) 'REMOVE)
 NIL

(EPROGN PLAN)
 (ON 5 2)

(SCENE)
 (ON 5 2)
 (ON 3 TABLE)
 (ON 4 TABLE)
 (ON 6 TABLE)
 (ON 2 1)
 (ON 1 TABLE)
NIL
```

Le plan généré correspond à la succession des scènes suivantes :



Le programme est donc capable

1) de commander une main mécanique pour accomplir les actions du plan de façon cohérente

2) de construire un plan correspondant à la tâche donnée, puis de stocker ce plan pour usage ultérieur ce qui correspond à une forme spécifique d'apprentissage.

Nous envisageons d'utiliser cette possibilité pour la construction automatique de programmes simples CONNIVER et LISP.

CONNIVER nous paraît un langage tout à fait adéquat au projet de décrire par programme l'idée que nous nous faisons d'un programmeur débutant.

LISP CAB 500

C'est une spécificité du Département Informatique de PARIS VIII de mettre à la disposition en libre service de ses étudiants 6 ordinateurs CAB 500. Ces machines TRES anciennes possèdent cependant un excellent jeu d'instructions qui en font un excellent outil pédagogique. Toutefois, leur mémoire sur tambour les rend particulièrement lentes.

Harald WERTZ a tenté et réussi l'expérience d'implanter un interprète LISP complet sur les CAB 500, en minimant de façon très astucieuse le temps de traitement. Les étudiants ont ainsi accès à un LISP conversationnel très moderne, comprenant certaines primitives CONNIVER.

Le manuel de référence du LISP CAB 500 est donné en annexe.

SYSTEMES DE PRODUCTIONS

En liaison avec Richard M.YOUNG, de l'Ecole d'Intelligence Artificielle de l'Université d'Edinburgh, Pierre Louis NEUMANN a implémenté sur l'ordinateur T 1600 un interprète de productions.

Ce type de système a été expressement conçu pour décrire par des programmes les régularités dans le comportement d'un sujet au cours d'une expérience. Un système de production (PS) est normalement composé d'un ensemble de règles exprimant quelle sera l'action du sujet sous certaines conditions. Une règle (ou production) aura la forme canonique :

nom : CONDITION - - - > ACTION

L'ensemble de ces règles constitue la MEMOIRE A LONG TERME (LTM) du système : elle n'est pas un principe modifiable.

Tout au contraire la zone de manoeuvre du système : la MEMOIRE A COURT TERME du système (STM) contient un ensemble d'informations hautement modifiables.

La STM est une liste ordonnée d'expressions symboliques qui représentent le "SAVOIR" courant immédiatement accessible au programme.
L'activité du système est induite par les productions de la LTM, dont les conditions sont des filtres pour les expressions de la STM. Les expressions de la STM qui sont filtrées par les conditions sont amenées en tête de la STM, de même que les expressions que le système crée de lui-même. Une expression représente un "PAQUET" d'informations accessible en bloc.

L'ensemble des règles de la LTM détermine totalement le comportement du système. Chacune des règles sera déclenchée par le filtrage réuss de sa condition avec une expression de la STM. L'action correspondan pourra être par exemple l'addition ou la suppression d'une expressic la création d'une nouvelle expression, des lectures ou des impressic L'état de la STM sera ainsi modifié, ce qui permettra à d'autres règ d'être déclenchées.

L'exemple le plus simple de PS est la description donnée par Newell et Simon du comportement d'un termostat supposé conserver la température d'une pièce entre 28° et 30° :

TPO: (Température>28°)‸(Température<30°)--->(STOP)

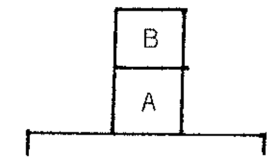
TP1: (Température<10°)--->(Appeler la maintenance)

TP2: (Température<28°)‸(FOYER=ETEINT)--->(Allumer-foyer)

TP3: (Température>30°)‸(FOYER=ALLUME)--->(Eteindre-foyer)

Voici un autre exemple :

Soit la STM décrivant une scène de bloc-ville :



STM: ((SUR A TABLE)(SUR B A))

LTM: (SUR ?X ₡Y)‸(SUR ₡Y ?Z)--->(ADD-TO-STM (SUR ₡X ₡Z))

La condition contient ici des variables. Leur valeur sera celle des constantes filtrées dans les expressions de la STM qui sera ainsi modifiée :

STM: ((SUR B TABLE)(SUR A TABLE)(SUR B A)

Ces systèmes ont été utilisés jusqu'ici pour analyser des protocoles. Les domaines d'activité auxquels les sujets avaient affaire étaient des problèmes simples de calcul propositionnel, des problèmes d'échecs et de crypto-arithmétique.

Nous avons l'intention d'utiliser ces systèmes pour l'apprentissage de la programmation : les diagnostics d'erreur livrés pouvant être très éclairants pour les débutants. Nous exploiteront également les systèmes de production pour décrire l'activité d'un débutant construisant un programme simple.

Le système de Pierre-Louis NEUMANN tient en mémoire en moins de 4K mots de 16 bits. NEUMANN travaille en ce moment aux procédures de communication PS-LISP.

PROGRAMMES DE JEUX

ECHECS

Patrick GREUSSAY a mis au point, en ALGOL CAE 510, puis en PL T T 1600, un programme capable de mener à bien la totalité d'une partie d'échecs.

La communication programme-partenaire s'effectue en notation d'échecs ordinaire et certaines facilitées sont implémentées qui permettent de

- . visualiser une position donnée
- . démarrer sur une position qui n'est pas la position initiale
- . obtenir certaines statistiques sur le comportement du programme, ainsi que le temps de calcul pour chaque coup
- . demander un conseil au programme si son partenaire est embarassé
- . règler la profondeur i.e. l'horizon du programme par demi-coups.

Le programme peut jouer blanc ou noir, indique les coups illégaux de son partenaire et déclare éventuellement le MAT s'il y a lieu.

Voici quelques échantillons de son comportement.

```
OPTIONS=0
PROFONDEUR=4
TEMPS=0
STATISTIQUES=
ECHIQUIER=
DN= N ---------> le programme joue blanc
P:E2-E4 ---------> son 1er coup
TEMPS=13 ---------> temps de calcul, en secondes.
?P:E7-E5# ---------> coup de l'adversaire
C:G1-F3
TEMPS=45
?C:B8-C6#
F:F1-B5
TEMPS=31
?XIMPRIM ---------> demande de visualisation : les pièces noires sont représentées par -P
                    et les cases vides par *
           -T  *  -F  -D  -R  -F  -C  -T
           -P  -P  -P  -P  *  -P  -P  -P
           *   *  -C   *   *   *   *   *
           *   F   *   *  -P   *   *   *
           *   *   *   *   P   *   *   *
           *   *   *   *   *   C   *   *
           P   P   P   P   *   P   P   P
           T   C   F   D   R   *   *   T
?XPREDIT ------> demande d'aide : << programme, que jouerais-tu à ma place ? >>
C:G8-F6
TEMPS=48
?C:G8-F6#
C:B1-C3                                        le programme
TEMPS=39                                       joue l'ouverture
?F:F8-E7#                                          espagnole.
R:0-0
TEMPS=33
?XIMPRIM

           -T  *  -F  -D  -R   *   *  -T
           -P  -P  -P  -P  -F  -P  -P  -P
           *   *  -C   *   *  -C   *   *
           *   F   *   *  -P   *   *   *
           *   *   *   *   P   *   *   *
           *   *   C   *   *   C   *   *
           P   P   P   P   *   P   P   P
           T   *   F   D   *   T   R   *

?R:0-0 #
F:B5*C6 ----> capture.
TEMPS=43
?P:D7*C6#
P:H2-H3
TEMPS=64
?
```

```
?XLEGAUX ----------▷ demande de liste des coups légaux
                     et de leur évaluation superficielle ([illegible])
?F: E7-C5#
13        P: D2-D4
11        C: F3xE5
10        C: F3-D4
8         C: C3-D5
4         T: F1-E1
3         R: G1-H1
2         P: G2-G4
2         P: D2-D3
2         P: B2-B4
2         P: A2-A4
2         R: G1-H2
2         D: D1-E2
2         T: A1-B1
1         P: H3-H4
1         P: G2-G3
1         P: B2-B3
1         P: A2-A3
0         D: D1-E1
-2        C: F3-G5
-4        C: C3-E2
-4        C: C3-B5
-10       C: F3-E1
-10       C: F3-H4
-12       C: C3-A4
-18       C: F3-H2
-20       C: C3-B1
T: F1-E1
TEMPS= 66
?T: F8-E8#
P: G2-G4
TEMPS= 61
?XIMPRIM

          -T   *   -F  -D  -T   *   -R   *
          -P  -P   -P   *   *  -P   -P  -P
           *   *   -P   *   *  -C    *   *
           *   *   -F   *  -P   *    *   *
           *   *    *   *   P   *    P   *
           *   *    C   *   *   C    *   P
           P   P    P   P   *   P    *   *
           T   *    F   D   T   *    R   *

?XPREDIR
T: E8-E6
TEMPS= 74
```

```
?XLEGAUX
?T:E5-E6#
24          P:D2-D4
11          C:F3*E5
10          C:F3-D4
10          T:E1-E3
8           C:C3-D5
6           P:D2-D3
6           P:B2-B4
6           P:A2-A4
6           T:E1-E2
4           R:G1-G2
3           P:H3-H4
3           P:B2-B3
3           P:A2-A3
3           R:G1-F1
2           R:G1-H2
2           D:D1-E2
2           T:A1-B1
1           R:G1-H1
0           P:G4-G5
0           T:E1-F1
-2          C:F3-G5
-4          C:C3-E2
-4          C:C3-B5
-10         C:F3-H4
-12         C:C3-A4
-18         C:F3-H2
-20         C:C3-B1
```

```
P:A2-A4
TEMPS=77
?P:H7-H6#
T:E1-E2
TEMPS=62
?P:B7-B5#
P:A4*B5
TEMPS=49
?P:C6*B5#
C:C3*B5
TEMPS=107
?P:A7-A5#
P:D2-D3
TEMPS=88
?XIMPRIM
```

```
-T   *  -F  -D   *   *  -R   *
 *   *  -P   *   *  -P  -P   *
 *   *   *   *  -T  -C   *  -P
-P   C  -F   *  -P   *   *   *
 *   *   *   *   P   *   P   *
 *   *   *   P   *   C   *   P
 *   P   P   *   T   P   *   *
 T   *   F   D   *   *   R   *
```

```
?P:C7-C6#
C:B5-C3
TEMPS=83
?P:A5-A4#
F:C1-E3
TEMPS=103
?T:A8-A5#
F:E3*C5
TEMPS=68
?T:A5*C5#
T:A1*A4
TEMPS=75
?XIMPRIM
```

```
 *   *  -F  -D   *   *  -R   *
 *   *   *   *   *  -P  -P   *
 *   *  -P   *  -T  -C   *  -P
 *   *  -T   *  -P   *   *   *
 T   *   *   *   P   *   P   *
 *   *   C   P   *   C   *   P
 *   P   P   *   T   P   *   *
 *   *   *   D   *   *   R   *
```

```
?T:E6-D6#
T:A4-A8
TEMPS=119
?D:D8-B6#
T:A8*C8
TEMPS=62
?T:D6-D8#
C:C3-A4
TEMPS=43
?D:B6-A5#
T:C8*D8
TEMPS=38
?D:A5*D8#
C:A4*C5
TEMPS=24
?D:D8-E7#
```

```
P:B2-B4
TEMPS=45
?D:E7-A7#
C:F3*E5
TEMPS=47
?D:A7-A3#
C:C5-D7
TEMPS=45
?C:F6*D7#
C:E5*E7
TEMPS=41
?D:A3*E4#
C:D7-E5
TEMPS=33
?P:C6-C5#
P:C2-C4
TEMPS=34
?D:B4-D3#
P:F2-F4
TEMPS=24
?P:F7-F6#
C:E5-G6
TEMPS=29
?IMPRIM
```

```
*  -D   *   *   *   *  -R   *
*   *   *   *   *   *  -P   *
*   *   *   *   *  -P   C  -P
*   *  -P   *   *   *   *   *
*   *   P   *   P   P   P   *
*   *   *   P   *   *   *   P
*   *   *   *   T   *   *   *
*   *   *   D   *   *   R   *
```

Pour un débutant, le programme joue raisonnablement bien son ouverture. Comme presque tous les programmes du même type, le programme développe une arborescence des coups légaux, affecte une valeur matérielle aux sommets terminaux et se décide pour son coup en fonction de la valeur remontée qui lui est la plus favorable.

Les limitations du programme sont claires : il est littéralement aveugle au delà de sa profondeur de prévision :

profondeur	signification
1	je joue ceci (se dit le programme)
2	je joue ceci et l'adversaire réplique cela
3	je joue ceci, l'adversaire réplique, et je joue
4	je joue, il réplique, et je répond, et il réplique.

A la profondeur 4 le programme évalue 200 000 positions en moyenne avec un temps de calcul de l'ordre de la minute.
Il est clair que les améliorations à porter au programme ne seront pas liées à la profondeur de la recherche.
Nous comptons y incorporer une bibliothèque d'ouvertures standard, principalement pour lui éviter de tomber dans les pièges d'ouvertures tendus par des joueurs d'expérience.

Le programme joue TRES MAL les fins de parties. Un programme tout à fait indépendant, obéissant à d'autres principes s'impose ici.

Dans sa forme actuelle, notre programme d'échecs est un bon programme de championnat CONTRE D'AUTRES PROGRAMMES D'ECHECS. Nous comptons bien le faire participer aux prochains tournois.

De plus, nous avons pu constater que le comportement d'un joueur humain subit de larges modifications face à son adversaire programmé : l'absence de composantes stratégiques est telle qu'elle semble inhiber celle de son partenaire humain. On assiste également à des cas de surestimation du comportement du programme, son partenaire lui attribuant certains raisonnements dont il est tout à fait incapable pour justifier certains de ses coups.

Nous mettons en ce moment au point un système de dialogue en français entre un programme et un partenaire humain A PROPOS d'une position. La langue des traités d'échecs est une des plus ambigües qui soit. Le système devra être capable de discuter le COMMENTAIRE d'une position donnée par son partenaire humain.

KALAH

Jérôme CHAILLOUX a implémenté en langage PL sur l'ordinateur T 1600 un programme jouant raisonnablement bien au Kalah.
Ce jeu semble, à des variantes près, très répandu en Afrique de l'Ouest, au Moyen Orient ainsi qu'en Indonésie.

Voici la position initiale de ce jeu à 2 joueurs A et B

	6	6	6	6	6	6		B
0							0	
	6	6	6	6	6	6		A

Chaque joueur possède 6 cases contenant 6 pions et un kalah vide.
Le gagnant est le joueur qui ramasse plus de la moitié des pions dans son kalah. Si toutes les cases d'un des joueurs sont vides les pions restant dans les cases de son adversaire sont alors placées dans son kalah et la partie se termine.
Chaque joueur procède ainsi : il ramasse les pions d'une de ses cases et les répartit un à un dans le sens opposé aux aiguilles d'une montre, dans chaque case rencontrée et dans son kalah, en omettant le kalah de son adversaire. Si le dernier pion attéri dans son propre kalah, ce joueur rejoue de nouveau. Si ce pion attéri dans une de ses cases vides, la case opposée de son adversaire est vidée de son contenu au profit du kalah du joueur, ainsi que le pion.

Voici à titre d'illustration, le début d'une partie :

```
    6  6  6  6  6  6
 0                    0   le joueur A joue
    6  6  6  6  6  6

    6  7  7  7  7  7
 0                    1          B
    6  6  6  6  6  0

    0  7  7  7  7  7
 1                    1          A
    7  7  7  7  7  0

    0  7  7  7  7  8
 1                    2          B
    0  8  8  8  8  1

                                 etc
```

Le programme utilise la procédure de MINIMAX ALPHA BETA et une fonction d'évaluation fort simple, qui lui donne une efficacité redoutable dès que la profondeur s'accroît. Toutefois la particularité du jeu est la possibilité de "reprise", i.e. rejouer plusieurs coups à la suite. Les procédures standard de minimax $\alpha-\beta$ ne prévoient pas ordinairement ce cas.

Jean-Eric SCHOETTL et Harald WERTZ ont mis au point une procédure universelle d'$\alpha-\beta$ remarquablement simple et rapide, qui inclut le cas précédent ainsi que la possibilité de parties nulles, ce qui, dans le programme d'échecs rend beaucoup plus aisés la détection et le traitement du pat.

LOGO-LISP

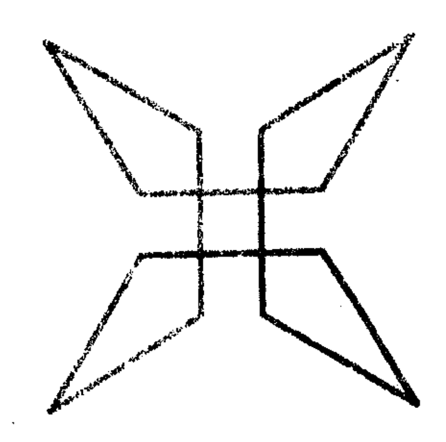
Nous envisageons la possibilité d'aborder l'initiation à la programmation sous forme graphique. LISP T 1600 comprenant des primitives de manipulation de table traçante, Patrick GREUSSAY a implémenté une version graphique du système de TORTUES du langage LOGO.

Le langage graphique est fort simple et comprend les primitives que voici :

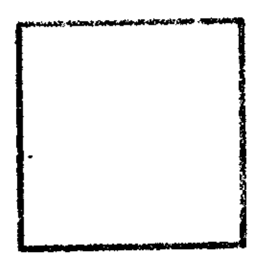
(FORWARD n)	:	n pas en avant
(BACKWARD n)	:	n pas en arrière
(RIGHT n)	:	dans la position courante de la plume, pivoter de n degrés à droite
(LEFT n)	:	pivoter de n degrés à gauche
(PENDOWN)	:	baisser la plume
(PENUP)	:	lever la plume

Voici dans les pages qui suivent des échantillons de résultats très variés d'appels des fonctions LISP-LOGO que voici :

```
(DE POLY (STEP ANGLE)
    (ESCAPE EXIT
            (WHILE T (TEST)
                   (FORWARD STEP)
                   (LEFT ANGLE)
)))
(DE POLYSPI (STEP ANGLE DECAL)
    (ESCAPE EXIT
            (WHILE T (TEST)
                   (FORWARD STEP)
                   (LEFT ANGLE)
                   (SETQ STEP (+ STEP DECAL))
)))
```



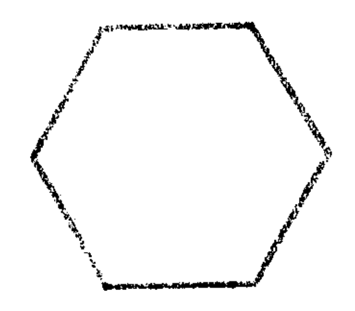
(POLY 90 30)



(POLY 30 90)



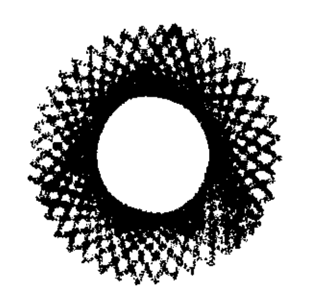
(POLY 20 120)



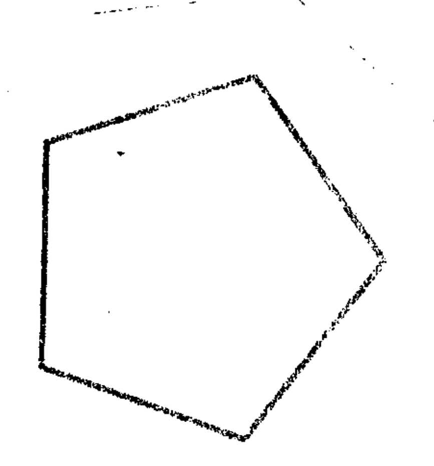
(POLY 20 60)



(POLY 10 123)

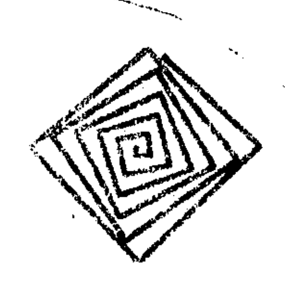


(POLY 30 123)



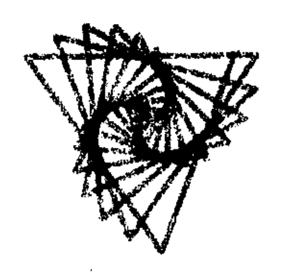
(POLY 30 72)



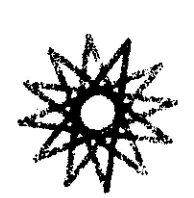


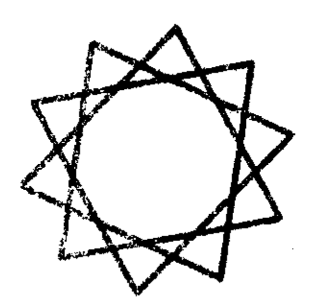
(POLYSPI 1 90 1)

(POLYSPI 1 92 1)



(POLYSPI 1 120 1)





SYSTEMES QUESTIONS-REPONSES EN LANGUE NATURELLE

Patrick GREUSSAY a implémenté en LISP T 1600 le système MILISY de T. Moran, permettant à un utilisateur d'introduire des règles syntaxiques et sémantiques lui permettant de soutenir avec le système un dialogue concernant un sujet donné.

Le système est décrit dans le rapport de Jean Paul BOUDIER et Sonia CHARALAMBIDES donné en annexe.

Nous travaillons actuellement sur une application inédite du système aux COMMENTAIRES DE POSITION DE JEU D'ECHECS. Voici un exemple de phrases décrivant une position :

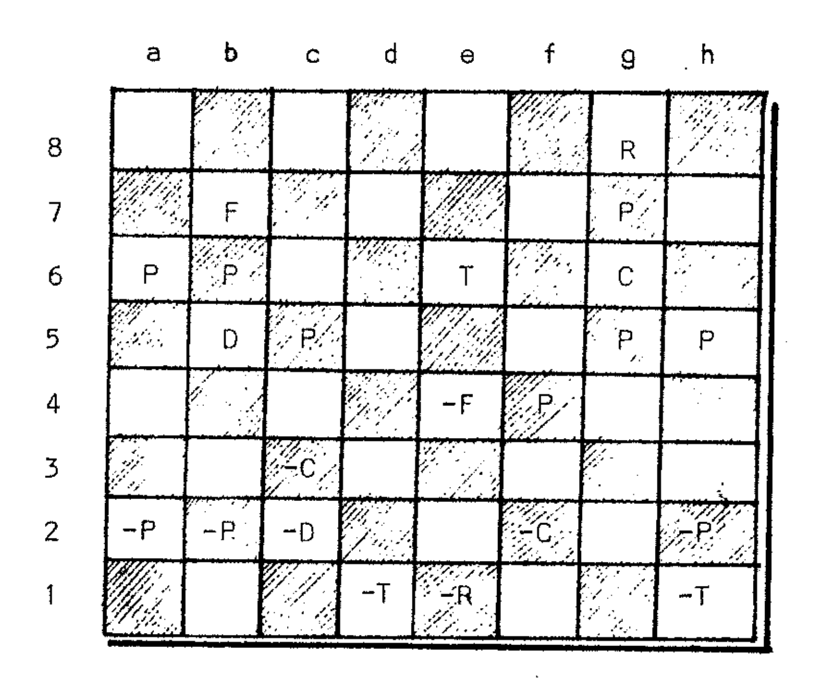
<<le cavalier qui défend le fou que la tour qui défend le cavalier cloue, attaque la dame>>

Il est clair que la phrase est ambigüe, en voici deux analyses :

(((C qui défend F) que T qui défend C cloue) attaque D)

(C qui défend (F que T qui défend C cloue) attaque D)

Lever l'ambigüité n'est ici possible que par rapport à un modèle i.e. la donnée d'une position.



A partir de la phrase donnée, le système doit pouvoir comprendre

1) (T1 défend C) C : cavalier quelconque

2) (T1 cloue F1)

3) (C1 défend F1)

4) (C1 attaque D)

et doit pouvoir répondre à des questions telles que par exemple

<<que déduire de la couleur des pièces?>>
ou
<<peut-on donner les coordonnées des pièces en question?>>

A moyen terme nous espérons mettre au point un programme-spectateur capable dans une mesure limitée de "comprendre" une partie d'échecs et d'en livrer en français un commentaire limité.